Deliverable 3
Team: The Textbook Triad
Project: UNTextbook Finder
Names: Jacob Cole, Parker Cantu, Michael Pittard

1. Introduction

1.1 This is the final deliverable that will contain a manual for the end-user, directions on how to compile/run the program and test suites, set of automated test cases, feature information, authors of components/classes, and meeting times.

2. Added Dependencies

2.1 Selenium[2]

To scrape the sites we needed to run the JavaScript associated with the page. In other words, we needed to scrape pages that dynamically changed based on user interaction. The Scrapy[1] framework we used didn't render JavaScript. So in order to have the JavaScript work, we had to use a library called Selenium. Specifically, we used Selenium[2] for Python. This allowed us to mimic browser clicks, hovers, etc. to have JavaScript react to such things. We also used the Selenium Standalone Server. This was used in order to use the HtmlUnitDriver from Selenium which is basically a browser engine.

2.2 Parse.com[5]

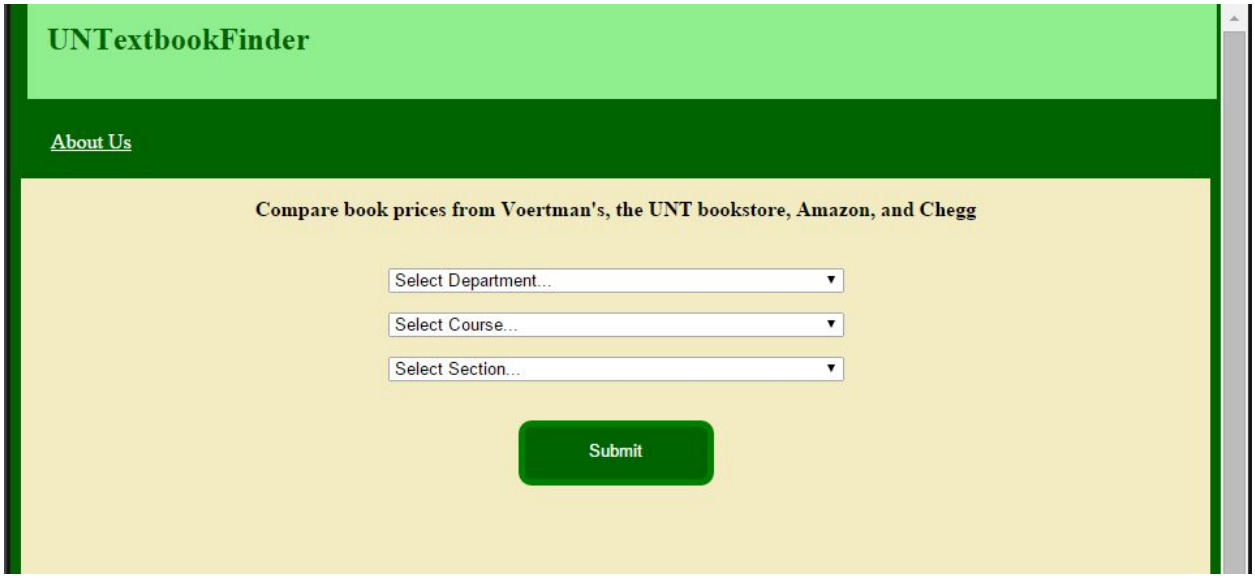
We did not have an easy way for all of us to access the database from a single source. Our fix was to use Parse.com. It's an online database that has many languages SDK's supported to obtain and create data. We used the PHP SDK, and Parse API wrapper written in Python called parse_rest.

3. Manual

Navigation

3.1 Homepage

After Navigating to the website using your web browser, the first page you should see is the homepage.



What you should see on the homepage.

Using the the three drop down menus on the homescreen, select the department of the book you will be searching for first. this will populate the second dropdown menu based on the department that you choose. For now, in the third drop down menu, just select “ALL”. This dropdown merely exists for a possible feature addition to the website. Click the “Submit” button to submit your search and load the results page.

3.2 Results Page

The results page shows you the textbook prices retrieved from our supported websites(UNT bookstore[4], and Voertman’s[3]) based on the class that you searched for. It takes a couple of seconds for the information to be scraped from the websites, so you may not see all of the information immediately. From here you can navigate back to the home page or to the about us page.

3.3 About Us Page

The about us page displays a small paragraph explaining what the project was made for and who made it.

4. Program Running / Compiling

4.1 Dependencies for testing

Our website requires the system you test this on to have Python 2.7, PHP 5.4+, and Java installed.

4.2 Compiling

This application will not require any compiling. It is a website that's written in Python, and PHP mainly. These are interpreted languages that don't require compiling.

4.3 Running

To run this website, you will need some sort of local webserver. XAMPP (or LAMPP for Linux) is the recommended webserver package as it's very easily setup. Assuming you are using XAMPP, copy our project folder to the 'htdocs' directory within the XAMPP directory. Now start Apache from the XAMPP control panel. Open up a terminal and navigate to the 'site' directory in our project. You will need to run 'java -jar selenium-server-standalone-2.48.2.jar' to start a dependency for web scraping. After you verify this is running, open up a browser and type in 'localhost/csce4444/site/' (csce4444 will be the name of the directory you copied into 'htdocs' so it could be different if you named the directory other than that). You should see our homepage.

4.4 Testing

After navigating to the 'site/siteTest/php' directory, you can run the unit test with 'php ../../phpunit-old.phar ParseHandlerTest.php'. This relative path is given so you don't have to set the PATH on your system.

5. Features

5.1 Implemented Features

- UNT bookstore scraper (core scraper run once a day)
- Voertman's Scraper
- Selection of course to find books
- Results of UNT bookstore and Voertman's book searched for
- Pull UNT bookstore data from our database (Parse.com)

5.2 Limitations

We were unable to implement the following features to our project:

- Amazon and Chegg book information
- Cart to keep the books you possibly want to checkout

5.3 Future Features / Improvements

We plan to add the rest of the features from our requirements. They are:

- Obtaining Amazon and Chegg book information
- Cart to hold the books you may want in case you want to search for others

New features planned:

- Advertisements for revenue
- User logins for easy state saving
- Use Amazon API instead of scraping for performance
- Option to exclude / include only certain stores

The following improvements are planned as well:

- Reducing the time Voertman's takes to scrape
- Some UI cleanup

6. Authors of Components

Jacob Cole: wrote the components that work with the Scrapy[1] framework and Selenium to retrieve the information from the third party sites, and the PHP to execute the Scrapy command and retrieve the data.
Parker Cantu: wrote the HTML, tags and CSS that formatted the website.
Michael Pittard: wrote the HTML for the dropdown lists' content and the controller.

7. Meeting Minutes

In total, the group spent a total of 1140 minutes in meetings together.

Meeting	Date	Time	Attendees	Location	Discussion
1	8/28/15	2:30-4:00pm	All	Discovery Park	Introductions
2	9/4/15	2:30-4:00pm	All	Discovery Park	Deliverable 1
3	9/6/15	3:00-6:00pm	All	Willis Library	Deliverable 1
4	9/10/15	8:00-9:00pm	All	Skype	Presentation
5	9/23/15	5:00-7:00pm	All	Skype	Deliverable 2
6	10/2/15	2:30-3:30pm	All	Discovery Park	Deliverable 2
7	10/4/15	8:00-10:00pm	All	Skype	Deliverable 2
8	10/30/15	2:30-4:00pm	All	Discovery Park	Project
9	11/6/15	2:30-4:00pm	All	Discovery Park	Project
10	11/13/15	2:30-6:00pm	All	Discovery Park	Project
11	11/18/15	8:30-10:00pm	All	Skype	Deliverable 3

References

1. Scrapy Framwork, http://scrapy.org
   The Python framework that we used to Scrape web pages. It provided us with an easy way to extract data from the websites we needed, and provide command line calls so PHP could easily interact with it.

2. Selenium, http://docs.seleniumhq.org
   The library that we used to render the JavaScript that was required to obtain the data from the scraped web pages. Used alongside the Scrapy framework.

3. Voertman's, https://voertmans.textbooktech.com
   The local book store in Denton that has a website for UNT students to select their books from.

4. UNT Bookstore, http://unt.bncollege.com
   The official bookstore of UNT partnered with Barnes and Noble. The site we used to obtain book data from and insert into our database.

5. Parse, https://parse.com
   The free, online solution for a central database that supports many language SDKs, and provides an API as well. We used this so we didn't have to set up a server and configure MySQL.